

Refine Search

Search Results -

Term	Documents
CLUSTER\$	0
CLUSTER	34569
CLUSTERA	1
CLUSTERABILITY	6
CLUSTERABLE	19
CLUSTERABLY	1
CLUSTERANALYSE	1
CLUSTERASSIGNED	1
CLUSTERATION	1
CLUSTERB	1
CLUSTERBALL	1
(L3 AND (CLUSTER\$ ADJ1 SWITCH\$)).USPT.	27

There are more results than shown above. [Click here to view the entire set.](#)

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Search:

L9

Search History

DATE: Thursday, March 25, 2004 [Printable Copy](#) [Create Case](#)

Set Name Query

side by side

Hit Count Set Name

result set

DB=USPT; PLUR=YES; OP=ADJ

L9 L3 and (cluster\$ adj1 switch\$)

27 L9

<u>L8</u>	L3 and (cluster\$ with switch\$)	75	<u>L8</u>
<u>L7</u>	L6 and (torus or wrap\$)	1	<u>L7</u>
<u>L6</u>	(5566342 or 5774698 or 5574939).pn.	3	<u>L6</u>
<u>L5</u>	(5566342 or 5774698 or 5574939).pn.	1	<u>L5</u>
<u>L4</u>	L3 and (cluster adj1 switch\$)	27	<u>L4</u>
<u>L3</u>	712/\$.cccls.	8535	<u>L3</u>
<u>L2</u>	(array adj1 processor\$).ab.	214	<u>L2</u>
<u>L1</u>	(array adj1 processor\$).pn.	0	<u>L1</u>

END OF SEARCH HISTORY

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KNAC	Draw Desc	Image
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	-----------	-------

☐ 2. Document ID: US 6023753 A

L6: Entry 2 of 2

File: USPT

Feb 8, 2000

DOCUMENT-IDENTIFIER: US 6023753 A
TITLE: Manifold array processor

US PATENT NO. (1):
6023753

Brief Summary Text (15):

To form an array in accordance with the present invention, processing elements may first be combined into clusters which capitalize on the communications requirements of single instruction multiple data ("SIMD") operations. Processing elements may then be grouped so that the elements of one cluster communicate within a cluster and with members of only two other clusters. Furthermore, each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By definition, in a SIMD torus with unidirectional communication capability, the North/South directions are mutually exclusive with the East/West directions. Processing element clusters are, as the name implies, groups of processors formed preferably in close physical proximity to one another. In an integrated circuit implementation, for example, the processing elements of a cluster preferably would be laid out as close to one another as possible, and preferably closer to one another than to any other processing element in the array. For example, an array corresponding to a conventional four by four torus array of processing elements may include four clusters of four elements each, with each cluster communicating only to the North and East with one other cluster and to the South and West with another cluster, or to the South and East with one other cluster and to the North and West with another cluster. By clustering PEs in this manner, communications paths between PE clusters may be shared, through multiplexing, thus substantially reducing the interconnection wiring required for the array.

CLAIMS:

1. An array processor, comprising:

N clusters wherein each cluster contains M processing elements, each processing element having a communications port through which the processing element transmits and receives data over a total of B wires;

communications paths which are less than or equal to (M)(B)-wires wide connected between pairs of said clusters; each cluster member in the pair containing processing elements which are torus nearest neighbors to processing elements in the other cluster of the pair, each path permitting communications between said cluster pairs in two mutually exclusive torus directions, that is, South and East or South and West or North and East or North and West; and

multiplexers connected to combine 2(M)(B)-wire wide communications into said less than or equal to (M)(B)-wires wide paths between said cluster pairs.

5. The array processor of claim 1, wherein a cluster switch comprises said multiplexers and said cluster switch is connected to mutlplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

10. An array processor, comprising:

N clusters wherein each cluster contains M processing elements, each processing element having a communications port through which the processing element transmits and receives data over a total of B wires and each processing element within a cluster being formed in closer physical proximity to other processing elements within a cluster than to processing elements outside the cluster;

communications paths which are less than or equal to (M) (B)-wires wide connected between pairs of said clusters, each cluster member in the pair containing processing elements which are torus nearest neighbors to processing elements in the other cluster of the pair, each path permitting communications between said cluster pairs in two mutually exclusive torus directions, that is, South and East or South and West or North and East or North and West; and

multiplexers connected to combine 2(M) (B)-wire wide communications into said less than or equal to (M) (B)-wires wide paths between said cluster pairs.

14. The array processor of claim 10, wherein a cluster switch comprises said multiplexer and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

28. A method of forming an array processor, comprising the steps of:

arranging processing elements in N clusters wherein each cluster contains M processing elements, such that each cluster includes processing elements which communicate only in mutually exclusive torus directions with the processing elements of at least one other cluster; and

multiplexing said mutually exclusive torus direction communications.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------

KWIC	Draw. Desc	Image
------	------------	-------

[Generate Collection](#)
[Print](#)

Term	Documents
MUTUALLY.USPT.	140445
MUTUALLIES	0
MUTUALLY	0
EXCLUSIVE.USPT.	82794
EXCLUSIVES.USPT.	24
TORUS.USPT.	4761
TORU.USPT.	7198
COMMUNICAT\$	0
COMMUNICAT.USPT.	10
COMMUNICATABILITY.USPT.	3
COMMUNICATABLE.USPT.	500
(L5 AND (COMMUNICAT\$ WITH MUTUALLY WITH EXCLUSIVE WITH TORUS)).USPT.	2

[There are more results than shown above. Click here to view the entire set.](#)

First Hit Fwd Refs

Generate Collection

L4: Entry 26 of 27

File: USPT

Nov 12, 1996

DOCUMENT-IDENTIFIER: US 5574939 A

TITLE: Multiprocessor coupling system with integrated compile and run time scheduling for parallelism

Abstract Text (1):

In a parallel data processing system, very long instruction words (VLIW) define operations able to be executed in parallel. The VLIWs corresponding to plural threads of computation are made available to the processing system simultaneously. Each processing unit pipeline includes a synchronizer stage for selecting one of the plural threads of computation for execution in that unit. The synchronizers allow the plural units to select operations from different thread instruction words such that execution of VLIWs is interleaved across the plural units. The processors are grouped in clusters of processors which share register files. Cluster outputs may be stored directly in register files of other clusters through a cluster switch.

Detailed Description Text (9):2. Cluster switch (C-Switch) 24,Detailed Description Text (18):

An M-Machine instruction consists of 12 operation fields, one for each operation unit, there being three operation units per cluster. Each cluster contains an integer operation unit, a floating-point operation unit and a memory interface operation unit, as well as an integer register file and a floating-point register file. Data may be transferred from one cluster 22 to another by writing a value directly into a remote register file via the Cluster Switch (C-Switch) 24. Special MOVE operations (IMOV, FMOV) are used to transfer data between clusters through the C-Switch thus avoiding traffic and lost transfer cycles through the M-Switch 34 and cache banks 28. The memory interface unit issues load and store requests to the memory system via the Memory Switch (M-Switch) 34 which routes a request to the appropriate cache bank 28.

Detailed Description Text (20):

The Cluster Switch (C-Switch) 24 is a crossbar switch with four buses. Each bus is connected directly to both of the register files in a cluster. There are 13 possible sources that may drive a bus: four integer arithmetic units (one per cluster), four floating-point arithmetic units (one per cluster), four cache banks, and the external memory interface. Data requested by load instructions are transmitted to register files via the C-Switch. The C-Switch includes arbitration logic for these buses. The priority is fixed in hardware with the cache banks having the highest priority.

Detailed Description Text (84):

The Cluster Switch (C-Switch) 24 is used to transport data form one cluster to another. It consists of four buses, one for each cluster. Each integer and floating-point function unit is capable of driving each C-Switch bus. In addition, each cache bank as well as the external memory interface may write to registers using the C-Switch. The C-Switch performs arbitration to determine which units will be allowed to write the bus. For the arithmetic units in the clusters, arbitration is performed one cycle before the result arrives in the WB pipeline stage. The

scoreboard is updated at this time as well. This allows those operations using the result to issue and meet up with the result at the EX stage without additional delay. The cache banks also reserve the C-Switch resources one cycle before the data is delivered on a load operation. The cache optimistically makes this reservation before it has determined whether the access is a hit. If the access misses the cache, the result is cancelled and the issue of the consuming instruction is inhibited.

Current US Original Classification (1):

712/24

Current US Cross Reference Classification (1):

712/200

WEST

Generate Collection

Print

Search Results - Record(s) 1 through 2 of 2 returned.☐ 1. Document ID: US 6338129 B1

L6: Entry 1 of 2

File: USPT

Jan 8, 2002

DOCUMENT-IDENTIFIER: US 6338129 B1

TITLE: Manifold array processor

US PATENT NO. (1):
6338129Brief Summary Text (14):

To form an array in accordance with the present invention, processing elements may first be combined into clusters which capitalize on the communications requirements of single instruction multiple data ("SIMD") operations. Processing elements may then be grouped so that the elements of one cluster communicate within a cluster and with members of only two other clusters. Furthermore, each cluster's constituent processing elements communicate in only two mutually exclusive directions with the processing elements of each of the other clusters. By definition, in a SIMD torus with unidirectional communication capability, the North/South directions are mutually exclusive with the East/West directions. Processing element clusters are, as the name implies, groups of processors formed preferably in close physical proximity to one another. In an integrated circuit implementation, for example, the processing elements of a cluster preferably would be laid out as close to one another as possible, and preferably closer to one another than to any other processing element in the array. For example, an array corresponding to a conventional four by four torus array of processing elements may include four clusters of four elements each, with each cluster communicating only to the North and East with one other cluster and to the South and West with another cluster, or to the South and East with one other cluster and to the North and West with another cluster. By clustering PEs in this manner, communications paths between PE clusters may be shared, through multiplexing, thus substantially reducing the interconnection wiring required for the array.

CLAIMS:

13. An array processor, comprising:

a plurality of processing elements (PEs) grouped in clusters, with each cluster communicating with two other clusters in mutually exclusive directions, each PE having a single inter-PE communications port for communicating with other PEs, each of said ports having a single input and a single output;

inter-PE communications paths connecting said single inter-PE communications ports through controllably switched cluster switches; and

the controllably switched cluster switches to select mutually exclusive inter-PE connection paths for PE to PE communication and connect the plurality of PEs into a torus connected array.

Hit List



Search Results - Record(s) 1 through 2 of 2 returned.

☐ 1. Document ID: US 6338129 B1

L5: Entry 1 of 2

File: USPT

Jan 8, 2002

DOCUMENT-IDENTIFIER: US 6338129 B1

TITLE: Manifold array processor

Drawing Description Text (22):

FIG. 18 is a block diagram illustrating one of the clusters of the embodiment of FIG. 17, which illustrates in greater detail a cluster switch and its interface to the illustrated cluster;

Detailed Description Text (4):

The PEs may be single microprocessor chips that may be of a simple structure tailored for a specific application. Though not limited to the following description, a basic PE will be described to demonstrate the concepts involved. The basic structure of a PE 30 illustrating one suitable embodiment which may be utilized for each PE of the new PE array of the present invention is illustrated in FIG. 3A. For simplicity of illustration, interface logic and buffers are not shown. A broadcast instruction bus 31 is connected to receive dispatched instructions from a SIMD controller 29, and a data bus 32 is connected to receive data from memory 33 or another data source external to the PE 30. A register file storage medium 34 provides source operand data to execution units 36. An instruction decoder/controller 38 is connected to receive instructions through the broadcast instruction bus 31 and to provide control signals 21 to registers within the register file 34 which, in turn, provide their contents as operands via path 22 to the execution units 36. The execution units 36 receive control signals 23 from the instruction decoder/controller 38 and provide results via path 24 to the register file 34. The instruction decoder/controller 38 also provides cluster switch enable signals on an output the line 39 labeled Switch Enable. The function of cluster switches will be discussed in greater detail below in conjunction with the discussion of FIG. 18. Inter-PE communications of data or commands are received at receive input 37 labeled Receive and are transmitted from a transmit output 35 labeled Send.

Detailed Description Text (17):

In FIG. 15A, clusters 80, 82 and 84 are three PE clusters connected through cluster switches 86 and inter-cluster links 88 to one another. To understand how the manifold array PEs connect to one another to create a particular topology, the connection view from a PE must be changed from that of a single PE to that of the PE as a member of a cluster of PEs. For a manifold array operating in a SIMD unidirectional communication environment, any PE requires only one transmit port and one receive port, independent of the number of connections between the PE and any of its directly attached neighborhood of PEs in the conventional torus. In general, for array communication patterns that cause no conflicts between communicating PEs, only one transmit and one receive port are required per PE,

independent of the number of neighborhood connections a particular topology may require of its PEs.

Detailed Description Text (18):

Four clusters, 44 through 50, of four PEs each are combined in the array of FIG. 15B. Cluster switches 86 and communication paths 88 connect the clusters in a manner explained in greater detail in the discussion of FIGS. 16, 17, and 18 below. Similarly, five clusters, 90 through 98, of five PEs each are combined in the array of FIG. 15C. In practice, the clusters 90-98 are placed as appropriate to ease integrated circuit layout and to reduce the length of the longest inter-cluster connection. FIG. 15D illustrates a manifold array of six clusters, 99, 100, 101, 102, 104, and 106, having six PEs each. Since communication paths 86 in the new manifold array are between clusters, the wraparound connection problem of the conventional torus array is eliminated. That is, no matter how large the array becomes, no interconnection path need be longer than the basic inter-cluster spacing illustrated by the connection paths 88. This is in contrast to wraparound connections of conventional torus arrays which must span the entire array.

Detailed Description Text (19):

The block diagram of FIG. 16 illustrates in greater detail a preferred embodiment of a four cluster, sixteen PE, manifold array. The clusters 44 through 50 are arranged, much as they would be in an integrated circuit layout, in a rectangle or square. The connection paths 88 and cluster switches are illustrated in greater detail in this figure. Connections to the South and East are multiplexed through the cluster switches 86 in order to reduce the number of connection lines between PEs. For example, the South connection between PE.sub.1,2 and PE.sub.2,2 is carried over a connection path 110, as is the East connection from PE.sub.2,1 to PE.sub.2,2. As noted above, each connection path, such as the connection path 110 may be a bit-serial path and, consequently, may be effected in an integrated circuit implementation by a single metallization line. Additionally, the connection paths are only enabled when the respective control line is asserted. These control lines can be generated by the instruction decoder/controller 38 of each PE.sub.3,0, illustrated in FIG. 3A. Alternatively, these control lines can be generated by an independent instruction decoder/controller that is included in each cluster switch. Since there are multiple PEs per switch, the multiple enable signals generated by each PE are compared to make sure they have the same value in order to ensure that no error has occurred and that all PEs are operating synchronously. That is, there is a control line associated with each noted direction path, N for North, S for South, E for East, and W for West. The signals on these lines enable the multiplexer to pass data on the associated data path through the multiplexer to the connected PE. When the control signals are not asserted the associated data paths are not enabled and data is not transferred along those paths through the multiplexer.

Detailed Description Text (20):

The block diagram of FIG. 17 illustrates in greater detail the interconnection paths 88 and switch clusters 86 which link the four clusters 44 through 50. In this figure, the West and North connections are added to the East and South connections illustrated in FIG. 16. Although, in this view, each processing element appears to have two input and two output ports, in the preferred embodiment another layer of multiplexing within the cluster switches brings the number of communications ports for each PE down to one for input and one for output. In a standard torus with four neighborhood transmit connections per PE and with unidirectional communications, that is, only one transmit direction enabled per PE, there are four multiplexer or gated circuit transmit paths required in each PE. A gated circuit may suitably include multiplexers, AND gates, tristate driver/receivers with enable and disable control signals, and other such interface enabling/disabling circuitry. This is due to the interconnection topology defined as part of the PE. The net result is that there are $4N_{sup.2}$ multiple transmit paths in the standard torus. In the manifold array, with equivalent connectivity and unlimited communications, only $2N_{sup.2}$

multiplexed or gated circuit transmit paths are required. This reduction of 2N.sup.2 transmit paths translates into a significant savings in integrated circuit real estate area, as the area consumed by the multiplexers and 2N.sup.2 transmit paths is significantly less than that consumed by 4N.sup.2 transmit paths.

Detailed Description Text (21):

A complete cluster switch 86 is illustrated in greater detail in the block diagram of FIG. 18. The North, South, East, and West outputs are as previously illustrated. Another layer of multiplexing 112 has been added to the cluster switch 86. This layer of multiplexing selects between East/South reception, labeled A, and North/West reception, labeled B, thereby reducing the communications port requirements of each PE to one receive port and one send port. Additionally, multiplexed connections between transpose PEs, PE.sub.1,3 and PE.sub.3,1, are effected through the intra-cluster transpose connections labeled T. When the T multiplexer enable signal for a particular multiplexer is asserted, communications from a transpose PE are received at the PE associated with the multiplexer. In the preferred embodiment, all clusters include transpose paths such as this between a PE and its transpose PE. These figures illustrate the overall connection scheme and are not intended to illustrate how a multi-layer integrated circuit implementation may accomplish the entirety of the routine array interconnections that would typically be made as a routine matter of design choice. As with any integrated circuit layout, the IC designer would analyze various tradeoffs in the process of laying out an actual IC implementation of an array in accordance with the present invention. For example, the cluster switch may be distributed within the PE cluster to reduce the wiring lengths of the numerous interfaces.

Detailed Description Text (22):

To demonstrate the equivalence to a torus array's communication capabilities and the ability to execute an image processing algorithm on the Manifold Array, a simple 2D convolution using a 3.times.3 window, FIG. 19A, will be described below. The Lee and Aggarwal algorithm for convolution on a torus machine will be used. See, S. Y. Lee and J. K. Aggarwal, Parallel 2D Convolution on a Mesh Connected Array Processor, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, pp. 590-594, July 1987. The internal structure of a basic PE 30, FIG. 3A, is used to demonstrate the convolution as executed on a 4.times.4 Manifold Array with 16 of these PEs. For purposes of this example, the Instruction Decoder/Controller also provides the Cluster Switch multiplexer Enable signals. Since there are multiple PEs per switch, the multiple enable signals are compared to be equal to ensure no error has occurred and all PEs are operating in synchronism. Based upon the S. Y. Lee and J. K. Aggarwal algorithm for convolution, the Manifold array would desirably be the size of the image, for example, an N.times.N array for a N.times.N image. Due to implementation issues it must be assumed that the array is smaller than N.times.N for large N. Assuming the array size is C.times.C, the image processing can be partitioned into multiple C.times.C blocks, taking into account the image block overlap required by the convolution window size. Various techniques can be used to handle the edge effects of the N.times.N image. For example, pixel replication can be used that effectively generates an (N+1).times.(N+1) array. It is noted that due to the simplicity of the processing required, a very small PE could be defined in an application specific implementation. Consequently, a large number of PEs could be placed in a Manifold Array organization on a chip thereby improving the efficiency of the convolution calculations for large image sizes.

CLAIMS:

1. An interconnection system for connecting a plurality of processing elements (PEs) in a torus-connected PE array, each PE having a communications port for communicating with the other PEs, the communications port including a single input and a single output, the interconnection system comprising:

inter-PE connection paths for connecting PEs grouped in clusters through cluster switches, with each cluster of PEs communicating with two other clusters of PEs in mutually exclusive directions through the cluster switches and inter-PE connection paths; and

the cluster switches connected to both the communications ports of said PEs and the inter-PE connection paths, and controllably switched to multiplex mutually exclusive communications onto the inter-PE connection paths connecting the cluster switches to reduce the number of communications paths required to provide inter-PE connectivity.

2. The interconnection system of claim 1, wherein a predetermined number of said plurality of PEs form pairs of transpose PEs, and wherein said cluster switches further comprise intra-cluster transpose connections to provide direct communications between the pairs of transpose PEs.

3. The interconnection system of claim 1, further comprising a control connected to the cluster switches for controlling the controllably switched cluster switches to select selectable modes of operation and wherein data and commands may be transmitted and received at said communications ports in one of four selectable modes:

a) a transmit east/receive west mode for transmitting data to an east PE via the communications port of the east PE while receiving data from a west PE via the communications port of the west PE;

b) a transmit north/receive south mode for transmitting data to a north PE via the communications port of the north PE while receiving data from a south PE via the communications port of the south PE;

c) a transmit south/receive north mode for transmitting data to a south PE via the communications port of the south PE while receiving data from a north PE via the communications port of the north PE; and

d) a transmit west/receive east mode for transmitting data to a west PE via the communications port of the west PE while receiving data from an east PE via the communications port of the east PE.

6. The interconnection system of claim 5, wherein said inter-PE connection paths are selectively switched through the cluster switches to select between different connection paths by paths enabling signals.

11. The interconnection system of claim 9, wherein the cluster switch supports an operation wherein the PEs are each simultaneously sending commands or data through the output while receiving commands or data through the input.

13. An array processor, comprising:

a plurality of processing elements (PEs) grouped in clusters, with each cluster communicating with two other clusters in mutually exclusive directions, each PE having a single inter-PE communications port for communicating with other PEs, each of said ports having a single input and a single output;

inter-PE communications paths connecting said single inter-PE communications ports through controllably switched cluster switches; and

the controllably switched cluster switches to select mutually exclusive inter-PE connection paths for PE to PE communication and connect the plurality of PEs into a torus connected array.

15. An array processor, comprising:

a plurality of processing elements (PEs) arranged in clusters, each each PE having a communications port for communicating with the other PEs, the communications port including a single input and a single output;

inter-PE communications paths connecting the PEs through cluster switches; and

the cluster switches operable to multiplex inter-PE communications and connect the PEs of each cluster for communication in mutually exclusive directions with the PEs of each of at least two other clusters utilizing the inter-PE communication paths.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Abstract	Claims	KWIC	Draw De
------	-------	----------	-------	--------	----------------	------	-----------	----------	--------	------	---------

☐ 2. Document ID: US 6023753 A

L5: Entry 2 of 2

File: USPT

Feb 8, 2000

DOCUMENT-IDENTIFIER: US 6023753 A

TITLE: Manifold array processor

Drawing Description Text (22):

FIG. 18 is a block diagram illustrating one of the clusters of the embodiment of FIG. 17, which illustrates in greater detail a cluster switch and its interface to the illustrated cluster;

Detailed Description Text (5):

The PEs may be single microprocessor chips that may be of a simple structure tailored for a specific application. Though not limited to the following description, a basic PE will be described to demonstrate the concepts involved. The basic structure of a PE 30 illustrating one suitable embodiment which may be utilized for each PE of the new PE array of the present invention is illustrated in FIG. 3A. For simplicity of illustration, interface logic and buffers are not shown. A broadcast instruction bus 31 is connected to receive dispatched instructions from a SIMD controller 29, and a data bus 32 is connected to receive data from memory 33 or another data source external to the PE 30. A register file storage medium 34 provides source operand data to execution units 36. An instruction decoder/controller 38 is connected to receive instructions through the broadcast instruction bus 31 and to provide control signals 21 to registers within the register file 34 which, in turn, provide their contents as operands via path 22 to the execution units 36. The execution units 36 receive control signals 23 from the instruction decoder/controller 38 and provide results via path 24 to the register file 34. The instruction decoder/controller 38 also provides cluster switch enable signals on an output the line 39 labeled Switch Enable. The function of cluster switches will be discussed in greater detail below in conjunction with the discussion of FIG. 18. Inter-PE communications of data or commands are received at receive input 37 labeled Receive and are transmitted from a transmit output 35 labeled Send.

Detailed Description Text (18):

In FIG. 15A, clusters 80, 82 and 84 are three PE clusters connected through cluster switches 86 and inter-cluster links 88 to one another. To understand how the manifold array PEs connect to one another to create a particular topology, the connection view from a PE must be changed from that of a single PE to that of the PE as a member of a cluster of PEs. For a manifold array operating in a SIMD

unidirectional communication environment, any PE requires only one transmit port and one receive port, independent of the number of connections between the PE and any of its directly attached neighborhood of PEs in the conventional torus. In general, for array communication patterns that cause no conflicts between communicating PEs, only one transmit and one receive port are required per PE, independent of the number of neighborhood connections a particular topology may require of its PEs.

Detailed Description Text (19):

Four clusters, 44 through 50, of four PEs each are combined in the array of FIG. 15B. Cluster switches 86 and communication paths 88 connect the clusters in a manner explained in greater detail in the discussion of FIGS. 16, 17, and 18 below. Similarly, five clusters, 90 through 98, of five PEs each are combined in the array of FIG. 15C. In practice, the clusters 90-98 are placed as appropriate to ease integrated circuit layout and to reduce the length of the longest inter-cluster connection. FIG. 15D illustrates a manifold array of six clusters, 99, 100, 101, 102, 104, and 106, having six PEs each. Since communication paths 86 in the new manifold array are between clusters, the wraparound connection problem of the conventional torus array is eliminated. That is, no matter how large the array becomes, no interconnection path need be longer than the basic inter-cluster spacing illustrated by the connection paths 88. This is in contrast to wraparound connections of conventional torus arrays which must span the entire array.

Detailed Description Text (20):

The block diagram of FIG. 16 illustrates in greater detail a preferred embodiment of a four cluster, sixteen PE, manifold array. The clusters 44 through 50 are arranged, much as they would be in an integrated circuit layout, in a rectangle or square. The connection paths 88 and cluster switches are illustrated in greater detail in this figure. Connections to the South and East are multiplexed through the cluster switches 86 in order to reduce the number of connection lines between PEs. For example, the South connection between PE.sub.1,2 and PE.sub.2,2 is carried over a connection path 110, as is the East connection from PE.sub.2,1 to PE.sub.2,2. As noted above, each connection path, such as the connection path 110 may be a bit-serial path and, consequently, may be effected in an integrated circuit implementation by a single metallization line. Additionally, the connection paths are only enabled when the respective control line is asserted. These control lines can be generated by the instruction decoder/controller 38 of each PE.sub.3,0, illustrated in FIG. 3A. Alternatively, these control lines can be generated by an independent instruction decoder/controller that is included in each cluster switch. Since there are multiple PEs per switch, the multiple enable signals generated by each PE are compared to make sure they have the same value in order to ensure that no error has occurred and that all PEs are operating synchronously. That is, there is a control line associated with each noted direction path, N for North, S for South, E for East, and W for West. The signals on these lines enable the multiplexer to pass data on the associated data path through the multiplexer to the connected PE. When the control signals are not asserted the associated data paths are not enabled and data is not transferred along those paths through the multiplexer.

Detailed Description Text (21):

The block diagram of FIG. 17 illustrates in greater detail the interconnection paths 88 and switch clusters 86 which link the four clusters 44 through 50. In this figure, the West and North connections are added to the East and South connections illustrated in FIG. 16. Although, in this view, each processing element appears to have two input and two output ports, in the preferred embodiment another layer of multiplexing within the cluster switches brings the number of communications ports for each PE down to one for input and one for output. In a standard torus with four neighborhood transmit connections per PE and with unidirectional communications, that is, only one transmit direction enabled per PE, there are four multiplexer or gated circuit transmit paths required in each PE. A gated circuit may suitably

include multiplexers, AND gates, tristate driver/receivers with enable and disable control signals, and other such interface enabling/disabling circuitry. This is due to the interconnection topology defined as part of the PE. The net result is that there are $4N_{sup.2}$ multiple transmit paths in the standard torus. In the manifold array, with equivalent connectivity and unlimited communications, only $2N_{sup.2}$ multiplexed or gated circuit transmit paths are required. This reduction of $2N_{sup.2}$ transmit paths translates into a significant savings in integrated circuit real estate area, as the area consumed by the multiplexers and $2N_{sup.2}$ transmit paths is significantly less than that consumed by $4N_{sup.2}$ transmit paths.

Detailed Description Text (22):

A complete cluster switch 86 is illustrated in greater detail in the block diagram of FIG. 18. The North, South, East, and West outputs are as previously illustrated. Another layer of multiplexing 112 has been added to the cluster switch 86. This layer of multiplexing selects between East/South reception, labeled A, and North/West reception, labeled B, thereby reducing the communications port requirements of each PE to one receive port and one send port. Additionally, multiplexed connections between transpose PEs, PE.sub.1,3 and PE.sub.3,1, are effected through the intra-cluster transpose connections labeled T. When the T multiplexer enable signal for a particular multiplexer is asserted, communications from a transpose PE are received at the PE associated with the multiplexer. In the preferred embodiment, all clusters include transpose paths such as this between a PE and its transpose PE. These figures illustrate the overall connection scheme and are not intended to illustrate how a multi-layer integrated circuit implementation may accomplish the entirety of the routine array interconnections that would typically be made as a routine matter of design choice. As with any integrated circuit layout, the IC designer would analyze various tradeoffs in the process of laying out an actual IC implementation of an array in accordance with the present invention. For example, the cluster switch may be distributed within the PE cluster to reduce the wiring lengths of the numerous interfaces.

Detailed Description Text (23):

To demonstrate the equivalence to a torus array's communication capabilities and the ability to execute an image processing algorithm on the Manifold Array, a simple 2D convolution using a $3_{times.3}$ window, FIG. 19A, will be described below. The Lee and Aggarwal algorithm for convolution on a torus machine will be used. See, S. Y. Lee and J. K. Aggarwal, Parallel 2D Convolution on a Mesh Connected Array Processor, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, No. 4, pp. 590-594, July 1987. The internal structure of a basic PE 30, FIG. 3A, is used to demonstrate the convolution as executed on a $4_{times.4}$ Manifold Array with 16 of these PEs. For purposes of this example, the Instruction Decoder/Controller also provides the Cluster Switch multiplexer Enable signals. Since there are multiple PEs per switch, the multiple enable signals are compared to be equal to ensure no error has occurred and all PEs are operating in synchronism. Based upon the S. Y. Lee and J. K. Aggarwal algorithm for convolution, the Manifold array would desirably be the size of the image, for example, an $N_{times.N}$ array for a $N_{times.N}$ image. Due to implementation issues it must be assumed that the array is smaller than $N_{times.N}$ for large N. Assuming the array size is $C_{times.C}$, the image processing can be partitioned into multiple $C_{times.C}$ blocks, taking into account the image block overlap required by the convolution window size. Various techniques can be used to handle the edge effects of the $N_{times.N}$ image. For example, pixel replication can be used that effectively generates an $(N+1)_{times.(N+1)}$ array. It is noted that due to the simplicity of the processing required, a very small PE could be defined in an application specific implementation. Consequently, a large number of PEs could be placed in a Manifold Array organization on a chip thereby improving the efficiency of the convolution calculations for large image sizes.

CLAIMS:

5. The array processor of claim 1, wherein a cluster switch comprises said multiplexers and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

6. The array processor of claim 5, wherein said cluster switch is connected to multiplex communications from the processing elements within a cluster for transmission to another cluster.

7. The array processor of claim 6, wherein said cluster switch is connected to multiplex communications between transpose processing elements within a cluster.

14. The array processor of claim 10, wherein a cluster switch comprises said multiplexer and said cluster switch is connected to mutliplex communications received from two mutually exclusive torus directions to processing elements within a cluster.

15. The array processor of claim 14 wherein said cluster switch is connected to multiplex communications from the processing elements within a cluster for transmission to another cluster.

16. The array processor of claim 15, wherein said cluster switch is connected to multiplex communications between transpose processing elements within a cluster.

25. An array processor, comprising:

processing elements (PEs) PE.sub.i,j, where i and j refer to the respective row and column PE positions within a conventional torus-connected array, and where $i=0,1,2,\dots,N-1$ and $j=0,1,2,\dots,N-1$, said PEs arranged in clusters PE.sub.(i+a) (ModN), (j+N-a) (ModN), for any i,j and for all a .epsilon. {0, 1, . . . , N-1}, wherein each cluster contains an equal number of PEs; and

cluster switches connected to multiplex inter-PE communications paths between said clusters thereby providing inter-PE connectivity equivalent to that of a torus-connected array.

26. The array processor of claim 25, wherein said cluster switches are further connected to provide direct communications between PEs in a transpose PE pair within a cluster.

Full	Title	Citation	Front	Review	Classification	Date	Reference	Sequences	Attachments	Claims	KIMC	Draw D
------	-------	----------	-------	--------	----------------	------	-----------	-----------	-------------	--------	------	--------

Clear	Generate Collection	Print	Fwd Refs	Back Refs	Generate OACS
-------	---------------------	-------	----------	-----------	---------------

Term	Documents
CLUSTER	34569
CLUSTRES	1
CLUSTRE	0
CLUSTERS	24923
SWITCH\$	0

SWITCH	550640
SWITCHA	3
SWITCHAABLE	1
SWITCHABE	2
SWITCHABILITY	219
SWITCHABILITYENTITY	1
(L1 AND (CLUSTER ADJ1 SWITCH\$)).USPT.	2

[There are more results than shown above. Click here to view the entire set.](#)

Display Format:

[Previous Page](#)

[Next Page](#)

[Go to Doc#](#)